

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.
530 Virginia Road, P.O. Box 9133
Concord, MA 01742-9133

Telephone: (978) 341-0036

Facsimile: (978) 341-0136

UNOFFICIAL DOCUMENT FOR EXAMINER'S REVIEW

FACSIMILE COVER SHEET

RECEIVED
CENTRAL FAX CENTER

JUN 22 2004

OFFICIAL

Examiner: Lechi Truong

Group: 2126

Date: June 22, 2004

Client Code: 1280

Facsimile No.: 703-872-9306

From: Ralph Tremontozzi

Subject: Paper: Amended Claims (draft)

Docket No.: 1280.2001-000

Applicants: Glen E. Salmon

Serial No.: 09/747,751

Filing Date: 12/22/00

Number of pages including this cover sheet: 8

Please confirm receipt of facsimile: Yes ☐ No ☐

Comments:

Privileged and Confidential - All information transmitted hereby is intended only for the use of the addressee(s) named above. If the reader of this message is not the intended recipient or the employee or agent responsible for delivering the message to the intended recipient(s), please note that any distribution or copying of this communication is strictly prohibited. Anyone who received this communication in error is asked to notify us immediately by telephone and to destroy the original message or return it to us at the above address via first class mail.

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.**Applicant Initiated Interview Request Form**Application No.: 09/747,751 First Named Applicant: Glen E. SalmonExaminer: Lechi Truong Art Unit: 2126 Status of Application: Pending

Tentative Participants:

(1) Examiner Lechi Truong (2) Examiner Meng-Ai An
 (3) Mary Lou Wakimura, Esq. (4) Ralph Tremontozzi, Esq.

Proposed Date of Interview: June 23, 2004 Proposed Time: 3:30 (PM)

Type of Interview Requested:

(1) ☒ Telephonic (2) ☐ Personal (3) ☐ Video Conference

Exhibit To Be Shown or Demonstrated: ☒ YES ☐ NOIf yes, provide brief description: Proposed amended claims 1, 9, 17, 18 and 19 (see attached)**Issues To Be Discussed**

Issues (Ref., Obj., etc.)	Claims / Fig. #s	Prior Art	Discussed	Agreed	Not Agreed
(1) Rejection	Claim 1	Dundon, Goldberg et al.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(2) Rejection	Claim 9	"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(3) Rejection	Claim 17	"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(4) Rejection	Claim 18	"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(5) Rejection	Claim 19	"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Brief Description of Arguments to be Presented:

Claims 1, 9, 17-19 include a proposed amendment to recite "in response to the received request" so as to distinguish over "loading and linking the required libraries before any API's are called" described in Dundon at col. 4, line 66 through col. 5, line 1. Thus, the proposed amended claims do "perform API mapping in response to an API being called" as described in the Office Action at hand at page 5, line 10.

An interview was conducted on the above-identified application on _____

NOTE: This form should be completed by applicant and submitted to the examiner in advance of the interview (see MPEP § 713.01).

This application will not be delayed from issue because of applicant's failure to submit a written record of this interview. Therefore, applicant is advised to file a statement of the substance of this interview (37 C.F.R. § 1.133(b)) as soon as possible.


 (Applicant/Applicant's Representative Signature)

 (Examiner/SPE Signature)

**RECEIVED
CENTRAL FAX CENTER****JUN 22 2004****DRAFT**

@PFDesktop\OL\MA\MHODMA\HBSR05\Manage\481889;1
1280.2001-000

OFFICIAL

1. (Currently amended): A method of automatically augmenting functionality in an application programming interface comprising the steps of:
 - receiving from an application a request for functionality to be fulfilled by a first component implementing at least a portion of the functionality within the application programming interface; and
 - in response to the received request:
 - querying the first component for the requested functionality;
 - receiving an indication from the first component that the requested functionality is not implemented by the first component;
 - searching for an augmentation component that implements the requested functionality;
 - loading the augmentation component to fulfill the request for functionality in place of the first component; and
 - responding to the request for functionality using the loaded augmentation component.
2. (Original): The method of Claim 1 wherein the searching step comprises looking for the augmentation component on a computer system executing the application.
3. (Original): The method of Claim 1 wherein the searching step comprises looking for the augmentation component on a network connected to a computer system executing the application.

DRAFT

4. (Original): The method of Claim 3 wherein the network uses a Uniform Resource Locator to address resources and the searching step uses the Uniform Resource Locator to direct the search.
5. (Original): The method of Claim 1 wherein the querying, receiving and searching steps create and maintain a catalog of available augmentation component functionality, the catalog being used to determine whether an augmentation component that implements the requested functionality exists.
6. (Original): The method of Claim 5 wherein the catalog is created upon the first request for functionality.
7. (Original): The method of Claim 5 wherein the catalog is recreated upon a specific request.
8. (Original): The method of Claim 1 wherein the first component is a connector for accessing information stored in databases, file systems, Enterprise Resource Planning systems or transaction systems.
9. (Currently amended): An apparatus for automatically augmenting functionality in an application programming interface comprising:
 - an application programming interface module receiving from an application a request for functionality to be fulfilled by a first component implementing at least a portion of the functionality within the application programming interface;
 - a query unit within the application programming interface module querying in response to the received request the first component for the requested functionality;

DRAFT

a receiver within the application programming interface module receiving an indication from the first component that the requested functionality is not implemented by the first component;

a search unit within the application programming interface module searching for an augmentation component that implements the requested functionality;

a loader within the application programming interface module loading the augmentation component to fulfill the request for functionality in place of the first component; and

a response unit within the application programming interface module responding to the request for functionality using the loaded augmentation component.

10. (Original): The apparatus of Claim 9 wherein the search unit comprises looking for the augmentation component on a computer system executing the application.
11. (Original): The apparatus of Claim 9 wherein the search unit comprises looking for the augmentation component on a network connected to a computer system executing the application.
12. (Original): The apparatus of Claim 11 wherein the network uses a Uniform Resource Locator to address resources and the searching unit uses the Uniform Resource Locator to direct the search.
13. (Original): The apparatus of Claim 9 wherein the receiver, the query unit, and the search unit create and maintain a catalog of available augmentation component functionality, the catalog being used to determine whether an augmentation component that implements the requested functionality exists.

DRAFT

14. (Original): The apparatus of Claim 13 wherein the catalog is created upon the first request for functionality.
15. (Original): The apparatus of Claim 13 wherein the catalog is recreated upon a specific request.
16. (Original): The apparatus of Claim 9 wherein the first component is a connector for accessing information stored in databases, file systems, Enterprise Resource Planning systems or transaction systems.
17. (Currently amended): An apparatus for automatically augmenting functionality in an application programming interface comprising:
 - a means for receiving from an application a request for functionality to be fulfilled by a first component implementing at least a portion of the functionality within the application programming interface;
 - a means for querying in response to the received request the first component for the requested functionality;
 - a means for receiving an indication from the first component that the requested functionality is not implemented by the first component;
 - a means for searching for an augmentation component that implements the requested functionality;
 - a means for loading the augmentation component to fulfill the request for functionality in place of the first component; and
 - a means for responding to the request for functionality using the loaded augmentation component.
18. (Currently amended): A computer program product comprising:
 - a computer usable medium for automatically augmenting functionality in an application programming interface;

DRAFT

a set of computer program instructions embodied on the computer usable medium, including instructions to:

receive from an application a request for functionality to be fulfilled by a first component implementing at least a portion of the functionality within the application programming interface; and

in response to the received request:

query the first component for the requested functionality;

receive an indication from the first component that the requested functionality is not implemented by the first component;

search for an augmentation component that implements the requested functionality;

load the augmentation component to fulfill the request for functionality in place of the first component; and

respond to the request for functionality using the loaded augmentation component.

19. (Currently amended): A computer data signal embodied in a carrier wave comprising a code segment for automatically augmenting functionality in an application programming interface, including instructions to:

receive from an application a request for functionality to be fulfilled by a first component implementing at least a portion of the functionality within the application programming interface; and

in response to the received request:

query the first component for the requested functionality;

receive an indication from the first component that the requested functionality is not implemented by the first component;

search for an augmentation component that implements the requested functionality;

DRAFT

load the augmentation component to fulfill the request for
functionality in place of the first component; and
respond to the request for functionality using the loaded
augmentation component.